文章编号:1671 - 6833(2013)05 - 0096 - 04

多 SP 轮函数对广义 Feistel 结构安全性影响分析

王田丽,黄 坤

(华北水利水电大学 数学与信息科学学院,河南 郑州 450046)

摘 要:广义 Feistel 结构使用多 SP 形式轮函数,会对安全性带来一定的影响.首次讨论了多 SP 轮函数在不完全扩散情况下,安全性发生的一些变化.并对 Lesamnta-256 的中间置换采用的结构和轮函数进行差分分析.这是长达三轮的超级匹配技术第一次用于广义 Feistel 结构的算法.

关键词:多 SP;Lesamnta-256;截断差分;反弹攻击;超级匹配

中图分类号:TP309 **文献标志码**:A doi:10.3969/j.issn.1671 - 6833.2013.05.021

0 引言

广义 Feistel 结构是由郑玉良等人于 1989 年的美洲密码年会上提出的一系列算法结构. Lesamnta[1]是进入 NIST 举办的 SHA-3 竞赛第一轮的 56 个杂凑函数之一,它采用了比较流行的 Type-1 广义 Feistel[2]结构的变体. 2010 年,Bouillaguet 等人对其给出了一个 20 轮的积分区分器[3]. 2009 年,Mendel 等人介绍了一种分析杂凑函数的新技术,称为反弹攻击[4]. 在 2011 年的 FSE 会议上,Sasaki 等人将这一技术用于 Feistel-SP 结构的算法,给出了 11 轮已知密钥区分器[5]. 后来,Sasaki 等人又将复杂度加以改进,并应用于 Camellia[6]. 但这种技术对广义 Feistel 结构进行攻击的结果如何,现在还不明晰. 笔者从多 SP 轮函数的角度出发,分析这种设计会给安全性带来什么样的影响.

1 基础知识

1.1 Type-1 广义 Feistel 结构和 Lesamnta 简介

Type-1 广义 Feistel 结构的状态有四个字,而 Lesamnta 的压缩函数利用分组密码和 MMO 结构,整个 Lesamnta 杂凑函数采用 MD 结构. 对于 Lesamnta-256 杂凑函数,消息填充之后长度是 256 比特的整数倍,并被分成 256 比特大小的块 $M_1 \| M_2 \| \cdots \| M_N$. 记压缩函数为 CF:$\{0,1\}^{256} \times \{0,1\}^{256} \to \{0,1\}^{256}$,它按照下列方式进行迭代:

$$H_i \leftarrow E_{H_{i-1}}(M_i) \oplus M_i, \quad i = 1,2,\cdots,N.$$

初始状态记为 H_0. 最后的 H_N 即为消息 M 的杂凑值. 中间的分组密码 E_K 为 4 分支的 Type-1 广义 Feistel 结构. 首先,中间链值 H_{i-1} 进入密钥生成算法,生成每一轮的 64 比特子密钥 k_j,其中 $0 \leqslant j \leqslant 31$. 状态中 4 个 64 比特的字我们用 W_j, X_j, Y_j, Z_j 来表示,则 E_K 的输出按照下列方式计算:

$$(W_0, X_0, Y_0, Z_0) \leftarrow M_i,$$

$$(W_{j+1}, X_{j+1}, Y_{j+1}, Z_{j+1}) \leftarrow (F(Y_j \oplus k_{j+1}) \oplus Z_j, W_{j+1}, X_{j+1}, Y_{j+1}),$$

$$0 \leqslant j \leqslant 31.$$

轮函数 F 包含 4 个运算:轮密钥加、字节代换、行移位层、列混淆层. 攻击是在已知密钥状态下,在这里忽略密钥扩展算法,细节参考文献[1].

1.2 反弹攻击

反弹攻击[4]是 Mendel 等人提出的一种针对杂凑函数的分析方法. 它从两个状态出发,选择截断差分,在中间某 S 盒层构造匹配,再分别向前向后构造截断差分路径,故命名为反弹攻击. 此类攻击的方法与传统的差分攻击有所不同,其关键点之一在于研究 S 盒层的匹配性质.

2 多 SP 轮函数 Type-1 广义 Feistel 结构安全性

利用三轮匹配技术,对两轮扩散(AES 类)的 SP 轮函数的 Type-1 广义 Feistel 结构进行分析.

收稿日期:2013 - 05 - 10;**修订日期**:2013 - 07 - 15

基金项目:国家自然科学基金资助项目(51190093)

作者简介:王田丽(1978 -),女,山东诸城人,华北水利水电大学讲师,硕士,从事信息安全方面研究,E-mail: wangtianli@ncwu.edu.cn.

情形 1：两轮扩散的双 SP 结构. 如果函数采用双 SP 结构，并且两轮达到全扩散，则可以采用如图 1 的弹入结构进行攻击. 图中 0 表示不活跃的字，F 表示全活跃的字，H 表示经过 P（或者 P^{-1}）达到全活跃的字. 首先在#A 处取一个活跃状态，此状态经过一个 P 扩散后全活跃，同样在#B 处选择一个活跃状态，经过一个 P^{-1} 扩散后全活跃，两个状态进行 3 轮匹配，如图 1 中粗线所示. 而#A 沿虚线运算经过一个逆 SP 层，然后异或轮子密钥，在第二轮异或常数，在第五轮异或轮子密钥，再进行双 SP 运算，如果设定第二轮的常数等于第一轮与第五轮轮密钥的异或，则可保证在第五轮异或的时候将差分消除，得到一个非活跃字.

弹出部分是概率为 1 的，具体每一轮的差分字特征见表 1. 弹出部分一共 11 轮，故攻击一共有 16 轮.

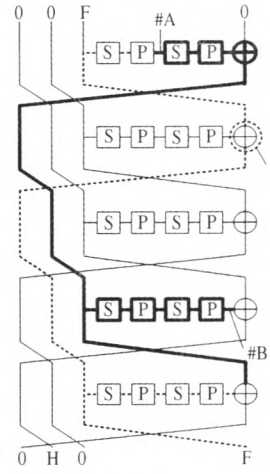

图 1　Type-1 双 SP 结构弹入攻击部分

Fig. 1　Inbound phase for Type-1 double SP constructure

表 1　Type-1 双 SP 结构与三 SP 结构弹出攻击部分

Tab. 1　Outbound phase for Type-1 double SP constructur and three SP structure

	Type-1 双 SP 结构弹入攻击部分					Type-1 三 SP 结构弹入攻击部分				
	轮次	状态				轮次	状态			
逆向弹出	5	0	F	F	F	5	0	F	F	F
	4	0	0	F	F	4	0	0	F	F
	3	0	0	0	F	3	0	0	0	F
	2	F	0	0	0	2	F	0	0	0
	1	0	F	0	0	1	0	F	0	0
	0	0	0	F	0	0	0	0	F	0
正向弹出	0	0	H	0	F	0	0	P(H)	0	F
	1	F	0	H	0	1	F	0	P(H)	0
	2	F	F	0	H	2	F	F	0	P(H)
	3	H	F	F	0	3	P(H)	F	F	0
	4	F	H	F	F	4	F	P(H)	F	F
	5	F	F	H	F	5	F	F	P(H)	F
	6	F	F	F	H	6	F	F	F	P(H)

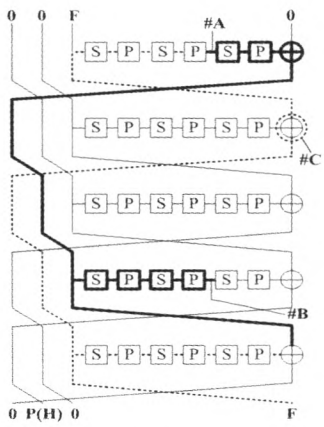

图 2　Type-1 三 SP 结构弹入攻击部分

Fig. 2　Inbound phase for Type-1 three SP constructure

情形 2：2 轮扩散的三 SP 结构. 这种轮函数使用了 3 轮的 SP 结构，这样看起来会使得扩散更加充分，但是总的攻击仍然可以达到 16 轮. 如图 2 所示，其弹出部分见表 1.

其它情形：由上述两种情形可以看到，即使再增加轮数，对攻击弹入部分的影响也只是将弹入五轮之后的第二个字变为 F，这样会将正向弹出部分轮数变为 3 轮，总的攻击轮数变为 13 轮.

3　Lesamnta-256 的截断差分区分器

3.1　弹入部分

步骤 1：（正向起始）选择并固定状态#4 的两个灰色字节的差分，此差分线性地传播至状态#5，

使得状态#5中有4个字节差分不为0.由于此状态第1行的每一个字节与其右下方的字节进入同一个超级S盒,将这两个字节看做一个整体,取遍其2^{16}个值,每个值均可以独立地计算至状态#9,然后存储在一个表中.这样,由状态#5至状态#9可以建立4个这样的表,每一个的大小为2^{16}.

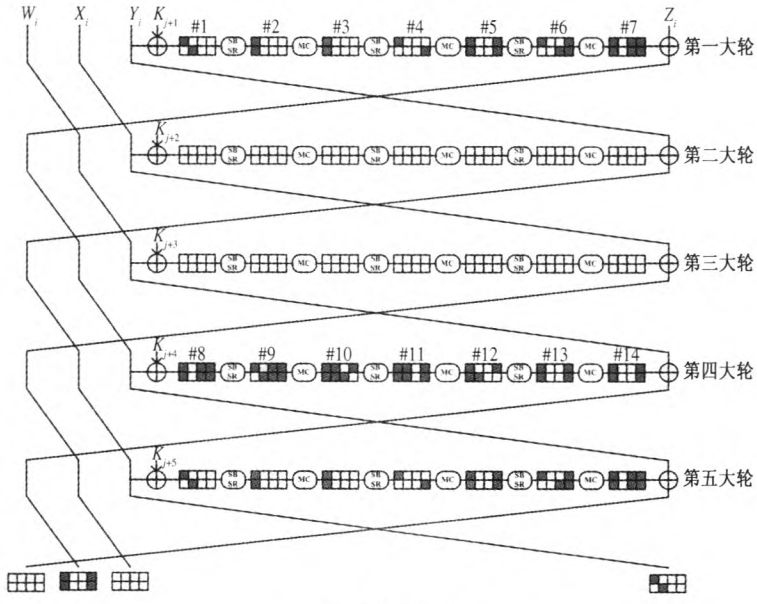

图3 弹入部分

Fig. 3 Inbound phase

步骤2:(逆向起始)相似地,在状态#12选择并固定4个灰色字节的差分,在状态#11取遍每一个超级S盒的输入的值,在状态#9建立4个大小为2^{16}的表.值得注意的是,在状态#9,由状态#5开始建立的每一个表都对应第1行的一个字节和它左下方的一个字节,由状态#11开始的每一个表都对应一列的两个字节.为了区别,我们从左到右记正向的四个表为l_1,l_2,l_3,l_4,逆向的四个表记为l_1',l_2',l_3',l_4'.

步骤3:(匹配轮)状态#9左下角字节差分为0,取定此字节的值.这样,相当于给集合l_1'限定了2^{16}个条件,而集合l_1'中包含2^{16}个元素,我们期望在其中找到一个满足条件.这样,状态#9和状态#10的第一行的两个字节的差分和值就都确定了.而随着状态#9左上角字节的差分和值的确定,相当于给集合l_1限定了2^{16}个条件,我们同样期望有一个元素满足条件.而集合l_1涉及的另一个字节为#9右下角的字节,所以这一字节的差分和值也确定了.这样,我们对集合l_4'限定了2^{16}个条件,我们期望从此集合中得到一个元素满足条件.按照同样的方法,我们可以确定剩下的字节的差分和值.状态#9中第1行的第2个字节是最后确定的元素.但是,我们发现此字节的差分为零,这样就在最后多出了2^8个条件.开始选择状态中左下角的元素值时,也有2^8个选择.这样,我们便以2^8的时间复杂度达到了匹配.

步骤4:(正向逆向弹出)找到匹配之后,状态#4逆向传播到状态#1,状态#12正向传播到状态#14.

步骤5:(消除轮)我们发现,第4大轮输出的最后一个字,和第5大轮F函数的输出都是由同样的截断差分模式生成的.而且,状态#1的差分与第5大轮第一个状态的差分相同,值相差一个常数.记第2大轮的F函数的输出为F_{i+2}^{out},则相差的常数可以表示为

$$K_{i+1}\oplus F_{i+2}^{out}\oplus K_{i+5}.$$

可以发现,F_{i+2}^{out}不含差分,是一个完全自由的字.所以,可以取$F_{i+2}^{out}=K_{i+1}\oplus K_{i+5}$.这样,便可以使得,第4大轮输出的最后一个字,和第5大轮F函数的输出的差分相同,通过异或消除.

通过上述步骤,我们得到了一条包含5大轮的差分路径.此路径的时间复杂度由两部分组成:建8个表的时间复杂度和中间匹配的时间复杂度.共用$2^{18.6}$次S盒的计算的时间,和2^{18}个"两字节"的存储.

3.2 弹出部分

事实上,弹出部分可以分为两个部分:正向弹出部分和逆向弹出部分.共有19轮,具体细节见表2.

表 2　19 轮截断差分路径

Tab.2　19 - round truncated differential path

	轮次	W_i	X_i	Y_i	Z_i		轮次	W_i	X_i	Y_i	Z_i
逆向弹出部分	0	6	8	8	8	正向弹出部分	13	2	0	4	0
	1	2	6	8	8		14	8	2	0	4
	2	0	2	6	8		15	4	8	2	0
	3	0	0	2	6		16	6	4	8	2
	4	0	0	0	2		17	8	6	4	8
	5	2	0	0	0		18	8	8	6	4
	6	0	2	0	0		19	8	8	8	6

3.3　与理想置换进行区分

用 $2^{18.6}$ 次 S 盒的计算的时间,2^{18} 个“两字节”的存储,构造了一条输入输出均有 2^{224} 种可能差分的差分路径.我们的攻击为 19 轮,每轮有 24 次的 S 盒计算,故时间复杂度为 $2^{18.6}/19 \times 24 = 2^{9.8}$.而状态含 32 个字节,故存储为 $2^{18}/32 = 2^{13}$.而一个理想置换,如果找到具有同样差分模式的两个输入/输出的话,需要 $2^{32/2} = 2^{16}$ 的计算复杂度.所以,此区分器可以将 19 轮的 Lesamnta 置换与理想函数有效区分.

4　结论

得到了 Type - 1 广义 Feistel-SPSP 结构的 16 轮已知密钥区分器,Type - 1 广义 Feistel-SPSPSP 结构的 16 轮已知密钥区分器.并证明了即使轮函数使用更多的 SP 迭代,也可以构造至少 13 轮的已知密钥区分器.还将这一方法应用与杂凑函数 Lesamnta,如何通过其构造碰撞或者近似碰撞值得进一步研究.

参考文献:

[1] SHOICHI H, HIDENORI K, HIROTAKA Y. SHA-3 Proposal: Lesamnta [EB/OL]. http://ehash.iaik.tugraz.at/uploads/5/5c/Lesamnta.pdf.

[2] ZHENG Yu-liang, MATSUMOTO T, IMAI H. On the construction of block ciphers provably secure and not relying on any unproved hypotheses [C]//Proceeding of CRYPTO. Santa Barbara, USA, 1989: 461 - 480.

[3] CHARLES B, ORR D, GAËTAN L P. Attacks on hash functions nased on generalized feistel: application to reduced-round lesamnta and SHAvite-3_{512} [C]// Proceeding of Selected Areas in Cryptography. Waterloo, Ontario, Canada, 2010: 18 - 35.

[4] MARIO L, FLORIAN M, CHRISTIAN R, et al. Rebound distinguishers: results on the full whirlpool compression function [C]//Proceeding of ASIACRYPT. Tokyo, Japan. 2009: 126 - 143.

[5] SASAKI Y. Meet-in-the-Middle Preimage Attacks on AES Hashing Modes and an Application to Whirlpool [C]//Proceeding of Fast Software Encryption. Lyngby, Denmark. 2011: 378 - 396.

[6] SASAKI Y, EMAMI S, HONG D, et al. Improved known-key distinguishers on feistel-SP ciphers and application to camellia [C]//Proceeding of Information Security and Privacy. Wollongong, NSW, Australia. 2012: 87 - 100.

Analysis of the Influence for a Generalized Feistel Network Using the Multi-SP Round Function

WANG Tian-li, HUANG Kun

(College of Mathematics and Information Sciences, North China University of Water Resources and Electric Power, Zhengzhou 450046, China)

Abstract: If a generalized Feistel network cipher adopts the multi-SP round function, the security is always different to the single-SP case. We studied this change where the multi-SP round function has a non-perfect diffusion. We analyzed the structure and the round function of the inner permutation of Lesamnta-256 for their differential properties. This is the first application of the super-match technique to a generalized Feistel algorithm.

Key words: multi-SP, Lesamnta-256, truncated differential, rebound attack, super-match